CLAIMS:

1.   In a compiler for generating an object
program from a source program, a compile method
comprising the steps of:

storing an object program, generated by
compiling a source program, in an object program file;
and

storing said source program corresponding to
said object program in said object program file, said
source program being associated with said object
program in said object program file.

2.   A compile method according to claim 1,
further comprising the steps of:

before generating an object program,
comparing said source program input to the compiler
with a source program corresponding to said object
program stored in an object program file;

detecting a changed portion in said source
program since the previous compiling;

compiling the changed portion of said input
source program; and

storing said object program generated by
compiling and the changed portion of said input source
program in said object program file.

3.   In a compiler for generating an object
program from a source program, a compile method
comprising the steps of:

obtaining source information by analyzing

syntax of a source program input to said compiler;

storing an object program generated by compiling said source information in said object program file; and

storing said source information corresponding to said object program in said object program file, said source information being associated with said object program in said object program file.

4.	A compile method according to claim 3, further comprising the steps of:

when generating said object program, comparing said source information obtained by analyzing syntax of said source program input to said compiler with said source information stored in said object program file;

detecting a change in said source information since previous compiling;

compiling the changed portion of said source information of the input source program; and

storing said object program and the changed portion of said source information in said object program file.

5.	A compile method according to claim 2, further comprising the steps of:

comparing said source programs on a procedure basis of said source program;

compiling said changed portion in said source program; and

storing said source program on a procedure unit basis in said object program file, said source program or said source information being associated with said object program in said object program file.

6.   A compile method according to claim 2, further comprising the steps of:

comparing said source programs on a processing step unit basis of said source programs;

compiling said changed portion in said source program; and

storing said source program on a processing step basis in said object program file, said source program or said source information being associated with said object program in said object program file.

7.   A compile method according to Claim 2, further comprising the steps of:

inputting an instruction specifying whether or not to compare said source program input to the compiler with said source program stored and associated with said object program in said object program file; and

if an instruction not to compare is input, compiling the entire said source program input to the compiler as said changed portion.

8.   A compile method according to claim 1, further comprising the steps of:

inputting an instruction specifying whether or not to store said source program in said object

program file; and

if an instruction not to store is input, storing said object program in said object program file.

9.　　　　A computer-readable recording medium having recorded therein a program for executing the compile method according to claim 1.

10.　　　A computer-readable recording medium having recorded thereon a program generated by the compile method according to claim 1.

11.　　　A compile method for generating an object program file from a source program having a plurality of procedures, comprising the steps of:

by regarding procedures of said source program as source-program compile units, compiling said source program on a procedure basis to generate said plurality of object-program compile units; and

in said object program file, storing said plurality of object-program compile units and said plurality of source-program compile units respectively associated with one another in said object program file, said plurality of source-program compile units being used to update said object program file on an object-program compile unit basis.

12.　　　A compile method according to claim 11, further comprising the steps of:

after making a change in said source program, inputting a source-program compile unit of a changed

source program;

out of said plurality of source-program compile units stored in said object program file, reading a source-program compile unit corresponding to said input source-program compile unit;

comparing said input source-program compile unit with said read source-program compile unit;

if the two of compile units do not coincide, compiling said input source-program compile unit to generate a new object-program compile unit;

in said object program file, updating said read source-program compile unit so as to be the same as said input source-program compile unit and updating the stored object-program compile unit corresponding to said read source-program compile unit to said new object-program compile unit.

13. A compile method according to claim 12, wherein even if the two of compile units do not coincide when said input source-program compile unit is compared with said read source-program compile unit, said input source-program compile unit is not compiled, if a content of a procedure of said input source-program compile unit is same as that of said read source-program compile units.

14. A compile method according to claim 11, further comprising the steps of:

in addition to storing said plurality of object-program compile units and said plurality of

source-program compile units, storing analysis information obtained by syntax analysis of said source program in said object program file;

after making a change in said source program, analyzing syntax of said source program;

comparing said analysis information obtained by said syntax analysis with said analysis information stored in said object program file;

if both the analysis information do not coincide, compiling a plurality of source-program compile units constituting said changed source program, to generate a plurality of new object-program compile units;

updating said plurality of source-program compile units stored in said object program file so as to be the same as said plurality of source-program compile units constituting said changed source program, updating said plurality of object-program compile units stored in said object program file so as to be the same as said new object-program compile units, and updating said analysis information stored in said object program file so as to be the same as the analysis information thus obtained.

15. A compile method according to claim 14, wherein said analysis information is a version of the compiler.

16. A compiler for generating an object program file from a source program having a plurality of

procedures, comprising:

a processing part for, by regarding procedures of said source program as source-program compile units, compiling said procedures on a procedure basis to generate a plurality of object-program compile units; and

a processing part for storing said plurality of object-program compile units and said plurality of ·source-program compile units. respectively associated with one another,

wherein said plurality of source-program compile units are used to update said object program file on an object-program compile unit basis.

17.   A compileraccording to claim 16, further comprising:

a processing part for, after making a change in said source program, inputting a source-program compile unit in said source program changed;

a processing part for reading a source-program compile unit corresponding to said input source-program compile unit out of said plurality of source-program compile units stored in said object program file;

a processing part for comparing said input source-program compile unit with said read source-program compile unit;

a processing part for, when both the compile units do not coincide, compiling said input source-

program compile unit to generate a new object-program
compile unit;

a processing part for, in said object program
file, updating said read source-program compile unit so
as to be the same as said input source-program compile
unit, and updating said stored object-program compile
unit corresponding to said read source-program compile
unit so as to be the same as said new object-program
compile unit.

18.   A compiler according to claim 16, further
comprising:

a processing part, in addition to storing
said plurality of object-program compile units and said
plurality of source-program compile units, storing
analysis information resulting from syntax analysis of
said source program in said object program file;

a processing part for analyzing syntax of
said source program after changes are made in said
source program;

a processing part for comparing analysis
information obtained by said syntax analysis with said
analysis information stored in said object program
file;

a processing part for, if both the analysis
information do not coincide, compiling a plurality of
source-program compile units constituting said changed
source program, to generate a plurality of new object-
program compile units; and

a processing part for updating said plurality
of source-program compile units stored in said object
program file so as to be the same as said plurality of
source-program compile units constituting said source
program changed, and updating said plurality of object-
program compile units stored in said object program
file so as to be the same as said plurality of new
object-program compile units, and updating said
analysis information stored in said object program file
so as to be the same as said analysis information
obtained.

19.   A compiler according to Claim 18, wherein
said analysis information is a version of the compiler.